

논문 2008-3-3

지능형 유비쿼터스 공간에서 상황 정보의 효과적인 저장 및 질의를 위한 정보 시스템의 설계 및 구현

Design and Implementation of an Information System for Storing and Querying Context Information in Ubiquitous Smart Space

이기혁*, 한형진*, 최원철*, 한경훈*, 한지연*, 손기락*

Ki-Hyeok Lee*, Hyung-Jin Han*, Won-Chul Choi*, Kyoung-Hoon Han*, Ji-Yean Han*, Kirack Sohn*

요 약

본 논문에서는 유비쿼터스 지능공간의 구현을 위해 CIB 프레임워크를 제안한다. 유비쿼터스 지능공간에서 유비쿼터스 모바일 디바이스는 주변 상황을 스스로 인지, 판단하여 서비스를 제공한다. CIB 프레임워크에서 유비쿼터스 모바일 디바이스는 리소스 및 정보의 효율적인 관리를 위해 RDF 형태의 저장방식을 가진다. 이 저장공간은 새로운 정보의 추가, 수정, 삭제에 유연하며, SPARQL을 이용하여 질의에 응답한다. 또한 주변 유비쿼터스 모바일 디바이스에 내부정보를 XML 형태로 제공하며, 외부에서 수집된 정보를 자신의 디바이스에 맞게 변환하여 저장하는 기능도 제공한다.

Abstract

This paper proposes a CIB framework implementing a ubiquitous smart space. In a ubiquitous smart space, a ubiquitous mobile device recognizes surrounding context, and infers the situation and provides services to users. Ubiquitous mobile devices have a RDF store to manage resources and relevant information efficiently on CIB framework. This store space is flexible on the operations such as addition, modification, deletion of information and response to query by SPARQL. In addition, it provides local information to other ubiquitous mobile devices in the form of XML and stores resource information from other ubiquitous mobile devices into local ubiquitous mobile device for the purpose of discovering external resources.

Key Words : Ubiquitous Computing, RDF, RDF Schema, Resource Sharing

1. 서 론

미래의 유비쿼터스 컴퓨팅은 컴퓨터 시스템이

“언제 어디서라도(anywhere, anytime)” 컴퓨팅 환경을 재구성할 수 있는 지능형 공간을 필요로 한다. 유비쿼터스 지능 공간(USS: Ubiquitous smart space)은 지능을 가진 오브젝트들을 환경에 내재시켜 시스템이 상황을 스스로 인지, 판단

*준회원, 한국외국어대학교 컴퓨터공학과
접수일자 : 2008.05.10, 수정완료일자 : 2008.6.16

하여 사용자에게 맞는 서비스를 적시에 자율적으로 제공하는 서비스공간이다 [1][2]. 이런 유비쿼터스 지능 공간에서의 모바일 오브젝트들을 본 논문에서는 U-MO (Ubiquitous Mobile Object) 라고 부르기로 한다. U-MO는 가지고 있는 각종 리소스들에 대한 정보를 저장하는 기능 및 저장된 정보에 대한 관리 및 질의 기능이 필요하다. 또한 내부 정보를 주변의 U-MO에게 제공이 가능하여야 한다. 이를 위하여 본 논문에서는 U-MO내에 자원 정보 및 객체 정보를 RDF [3] 형태로 저장한다. 각 U-MO는 이를 위한 저장 공간인 Context Information Base(CIB)를 구성한다. CIB는 리소스 및 정보를 효율적으로 저장할 수 있도록 RDF를 기반으로 한 모델로 구성되며, 이 저장 공간은 새로운 모델의 생성 및 정보의 추가, 삭제에 유연하다. 저장된 정보는 SPARQL[4]을 이용하여 질의에 응답한다. 또한 외부에서 수집된 정보를 자신의 저장 공간에 맞게 변환하는 기능과 내부 정보로부터 외부의 U-MO에 제공할 수 있도록 XML[5]형태의 정보를 생성하는 기능도 제공된다. 본 논문은 그림 1의 유비쿼터스 모바일 Software Platform에서 유비쿼터스 지능형 공간의 자원 정보 관리를 위한 CIB Framework에 대해 기술한다.

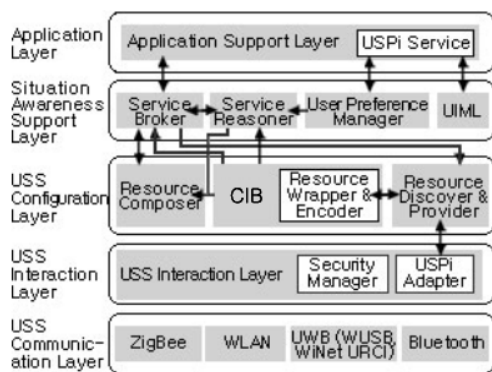


그림 1. 소프트웨어 플랫폼
Fig. 1. Software Platform

II. CIB Overview and Abstraction

1. CIB Overview

가. Context Information

U-MO가 사용할 수 있는 각종 리소스와 object에 대한 정보를 Context Information으로 정의한다. U-MO Software Platform 에서 context를 관리하는 모듈을 CIB(Context Information Base)라고 한다.

Context는 다음과 같은 정보를 의미한다.

- 하드웨어 Context: 프린터, 모니터, 키보드 등의 주변 device 리소스 정보
- 사용자 Context: 사용자 Profile, 위치 등의 사용자 정보
- 물리적 Context: 온도, 습도, 조도 등의 물리적 정보
- 히스토리 Context: 사용자의 Preference 정보., 시간에 흐름에 따른 Context Record

나. CIB Introduction

CIB는 U-MO가 사용할 수 있는 각종 Context Information들에 대한 정보를 저장하는 기능 및 저장된 정보에 대한 관리 및 질의 기능, 내부 context 정보를 이용하여 주변의 U-MO에게 제공할 수 있는 리소스들에 대한 세부 description 정보를 생성하는 기능 등을 제공한다.

CIB는 다음과 같은 리소스들에 대한 정보를 포함한다.

- 로컬 리소스 정보
- 공유 가능한 로컬 리소스 정보
- 주변 U-MO로부터 제공받은 리소스 정보

CIB는 U-MO Software Platform 상에서 Resource Discovery로부터 수집된 리소스 및 context 정보로부터 필요한 데이터를 추출하여 저장/관리하며, 저장된 context 정보는 Service Broker, Service Reasoner, Resource Composer, Resource Provider 가 동작하기 위해 필요한 정

보를 제공한다. CIB는 CIB storage, CIB IB Manager, Resource Wrapper & Encoder로 구성된다. CIB storage는 다단계의 category로 구성되는 리소스 및 서비스, 주변의 U-MO들이 제공하는 공유 리소스 및 context 정보들을 저장하는 기능을 수행한다. 효율적인 데이터 관리 및 질의 기능을 지원하기 위한 CIB IB Manager는 저장된 CIB에 대한 다양하고 편리한 방식의 질의 기능과 CIB에 대한 추가/삭제 등 관리 기능을 제공한다. Resource Wrapper는 내부 context 정보를 기반으로 외부 U-MO의 공유 리소스 정보를 변환 또는 추가하는 기능을 제공한다. U-MO가 자신이 제공할 수 있는 서비스에 대한 상세 정보를 외부의 U-MO에게 제공하기 위해 Resource Encoder를 이용하여 내부 context 정보를 기반으로 Resource description 정보 생성 할 수 있으며 생성된 정보는 Resource Provider/Discovery를 통해 다른 U-MO들에게 전달된다.

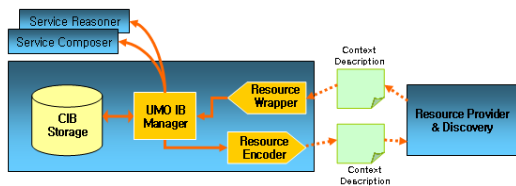


그림 2. CIB 구조
Fig. 2. CIB Structure

2. CIB Abstraction

가. CIB Abstraction

CIB는 Local U-MO 정보 및 주변 U-MO에 대한 상세 정보를 포함하며 각 U-MO 들이 제공하는 공유 리소스들에 대한 속성 및 부가기능에 대한 상세 기술 정보들을 포함한다. 또한 이러한 정보들은 Resource Discovery의 동작에 의해 동적으로 추가, 삭제 등의 변경 작업이 일어나며 다양한 형태의 리소스들에 대한 세부 정보를 표현할 수 있어야 한다. 또한 이러한 세부 정보에 대한 편리한 질의 방법이 제공되어야 한다.

이러한 다양한 표현력, 확장성에 대한 요구에 의해 CIB는 RDF형태로 저장되며 정보에 대한 질의는 SPARQL을 사용한다. CIB에 저장되는 정보의 형태는 RDF 스키마[6]로 정의된다.

RDF 스키마는 다음과 같은 클래스를 가진다.

- Local Resource: Local U-MO 정보 및 Local resource에 대한 정보를 기술하며, 다른 U-MO들에 의해 공유되어 이미 등록된 리소스 정보도 포함한다.
- Discovered Resource: Resource Discovery 프로세스에 의해 발견된 모든 U-MO 정보 및 device들이 제공하는 공유 리소스 정보들을 포함한다.
- Virtual Resource: 프로세스에 의해 발견된 모든 U-MO 정보 및 device들이 제공하는 공유 리소스 정보들을 포함한다.
- Device: Device에 대한 정의를 가진다. 각각의 Device는 자신이 로컬 리소스인지, Virtual Resource인지, Discovered Resource 인지를 나타낸다.
- UMO: U-MO에 대한 정보와, 해당 리소스에 대한 정보를 가진다.
- Public: 공통으로 가질 수 있는 정보를 나타낸다.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [<!ENTITY rdf
'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<ENTITY a 'http://protege.stanford.edu/system#'>
<ENTITY umo 'http://www.keti.re.kr/ubicom/umo#'>
<ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
xmlns:a="&a;"
xmlns:umo="&umo;"
xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&umo;class"
rdfs:label="class">
<rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&umo;Device"
rdfs:label="Device">
<rdfs:subClassOf rdf:resource="&umo;class"/>
</rdfs:Class>
<rdfs:Class rdf:about="&umo;DiscoveredResource"
```

```

    rdfs:label="DiscoveredResource">
    <rdfs:subClassOf rdf:resource="&umo:class"/>
</rdfs:Class>
<rdfs:Class rdf:about="&umo:LocalResource"
    rdfs:label="LocalResource">
    <rdfs:subClassOf rdf:resource="&umo:class"/>
</rdfs:Class>
<rdfs:Class rdf:about="&umo:UMO"
    rdfs:label="UMO">
    <rdfs:subClassOf rdf:resource="&umo:class"/>
</rdfs:Class>
<rdfs:Class rdf:about="&umo:VirtualResource"
    rdfs:label="VirtualResource">
    <rdfs:subClassOf rdf:resource="&umo:class"/>
</rdfs:Class>
<rdfs:Class rdf:about="&umo:Public"
    rdfs:label="struct">
    <rdfs:subClassOf rdf:resource="&umo:class"/>
</rdfs:Class>
</rdf:RDF>

```

위의 RDFS (RDF 스키마) 는 Local Resource, Discovered Resource, Virtual Resource, Device, UMO, Public클래스 사이의 관계를 RDF 스키마로 나타낸 것이다.

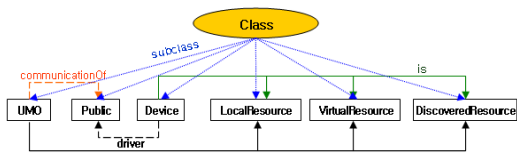


그림 3. RDF 스키마 그래프
Fig. 3. RDF Schema Graph

그림 3은 위의 RDFS를 RDF Graph로 나타낸 것이다. U-MO는 Local Resource, Virtual Resource, Discovered Resource를 포함한다. U-MO는 Public클래스의 정보를 이용해서 다른 U-MO와 communication을 하고, 모든 device들은 Local Resource, Virtual Resource, Discovered Resource 중에 하나 일 수 있다. 이 모든 항목들은 모두 최상위 클래스의 sub클래스이다.

나. UMO 클래스

UMO클래스는 U-MO device 의 고유 ID, 이름, vendor, version 등에 대한 상세 정보를 기술하며, 다음과 같은 Property를 포함한다.

- UMO_id: U-MO device에 부여된 고유한 ID 값을 포함한다.
- UMO_name: U-MO device에 부여된 이름을 표현한다. U-MO device의 이름은 임의로 부여될 수 있으며 유일성을 부여할 필요는 없다.
- UMO_Annotation: Annotation은 U-MO device에 대한 human-readable한 설명을 기술한다.
- UMO_Vendor: U-MO device의 제공자, 공급자에 대한 정보를 기술한다.
- UMO_Version: U-MO device의 version 정보를 제공한다.
- UMO_Agent_compliance: U-MO device가 U-MO Software Framework가 적용된 USS 공간안에서 활동하는 Agent인지를 확인하는 정보를 제공한다.
- isLocalResourceOf: U-MO device가 LocalResource인지를 나타낸다.
- isVirtualResourceOf: U-MO device가 VirtualResource인지를 나타낸다.
- isDiscoveredResourceOf: U-MO device가 DiscoveredResource인지를 나타낸다.
- HasLocalResource: U-MO device가 LocalResource정보를 가지고 있음을 나타낸다.
- HasVirtualResource: U-MO device가 VirtualResource정보를 가지고 있음을 나타낸다.
- HasDiscoveredResource: U-MO device가 DiscoveredResource정보를 가지고 있음을 나타낸다.

UMO클래스는 위와 같은 Property로 현재 U-MO의 기본적인 정보와, UMO클래스와 LocalResource 클래스, DiscoveredResource 클래스, VirtualResource 클래스들과의 관계를 나타낼 수 있다.

다. Device 클래스

Device클래스는 U-MO device 의 고유 ID, 이름, vendor, version 등에 대한 상세 정보를 기술하며, 다음과 같은 Property를 포함한다.

- isLocalResourceOf: Device가 LocalResource 인지를 나타낸다.
- isVirtualResourceOf: Device가 VirtualResource인지를 나타낸다.
- isDiscoveredResourceOf: Device가 DiscoveredResource인지를 나타낸다.

라. Public 클래스

Public 클래스는 U-MO가 공통적으로 가지거나 사용할 수 있는 Board, Network 등과 같은 정보를 나타낸다.

마. LocalResource 클래스

LocalResource 클래스는 U-MO platform을 통해 다른 U-MO device에 제공할 수 있는 자신의 리소스에 대한 정의를 기술한다. 리소스의 내용은 사용할 수 있는 device 정보, 디바이스 드라이버 정보, 해당 리소스에 대한 property 목록 및 부가적인 기능 목록 등이 있다.

바. VirtualResource 클래스

VirtualResource 클래스는 U-MO Software Platform을 통하여 다른 device로부터 제공받아 설치된 가상 리소스 정보임을 나타낸다.

사. DiscoveredResource 클래스

DiscoveredResource 클래스는 Resource Discovery를 통해 획득한 주변의 모든 사용 가능한 리소스들에 대한 상세 정보임을 나타낸다. DiscoveredResource 정보는 ResourceDiscovery가 주변 USS환경의 U-MO device의 정보를 탐색해 올 때마다, 기존의 정보는 삭제되고 새로 생성된다.

III. U-MO IB Manager

1. CIB Storage

U-MO IB (Information Base) Manager는 CIB에 관련된 정보를 저장, 추가, 변경, 삭제 및 질의에 대한 서비스를 제공하는 기능을 수행한다. U-MO Software Platform의 구성 요소들은 U-MO IB Manager를 통해 CIB 정보에 접근할 수 있다.

U-MO IB Manager는 그림 4와 같이 CIB Storage, Query Processor, Manipulation Processor, Resource Encoder 및 Resource Wrapper로 구성되어 있다.

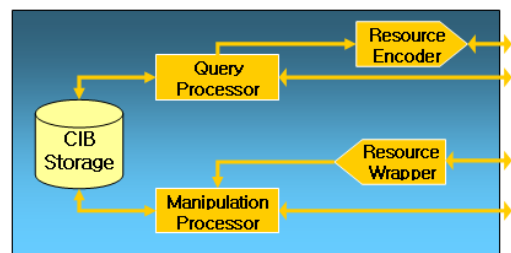


그림 4. U-MO IB Manager
Fig. 4. U-MO IB Manager

CIB Storage는 CIB 관련 정보를 물리적으로 저장 및 관리하기 위한 모듈로 RDF로 정의된 CIB 정보를 저장하고, 저장된 정보에 대한 검색을 위해 SPARQL 질의를 지원한다. CIB Storage는 Jena[7]를 통한 RDF모델의 생성, 추가, 수정 및 삭제 기능을 지원한다.

Jena는 Semantic Web application을 만들기 위한, 자바 프레임워크이다. Jena는 다음과 같은 기능을 제공한다.

- RDF로부터, RDF Model을 생성하여 모델의 추가 및 수정, 삭제를 지원한다.
- RDF를 검색, 수정, 추가 삭제를 위한 Query Language인 RDQL/SPARQL을 지원한다.
- RDF API를 제공한다.

- OWL API를 제공한다
- RDF를 RDF/XML, N3, N-Triples 형태로 읽고 쓰기를 지원한다.
- RDF데이터를 위한 In-memory / Persistent 스토리지를 제공한다.

2. Query Processor

Query Processor는 U-MO Software Platform의 구성요소들의 CIB 정보 검색 요청을 처리하여 CIB에 대한 질의를 수행하고 질의 결과를 반환하는 역할을 수행한다.

질의 요청은 다음과 같은 2가지의 질의 형태를 나타낸다.

- 공유 리소스 정보 요청
- CIB 정보에 대한 정보검색 요청

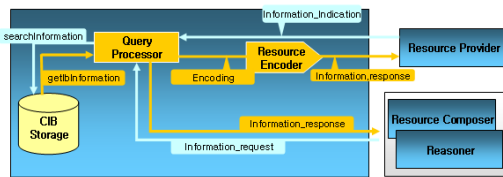


그림 5. Query Processor
Fig. 5. Query Processor

가. 공유 리소스 정보 요청

공유 리소스 정보 요청은 외부 U-MO로부터 resource discovery request를 받은 경우 Resource Provider에 의해 생성된다. Resource Provider는 XML형태의 Discovery Query의 요청을 U-MO IB Manager에게 전달하면 U-MO IB Manager의 Query Processor는 Discovery Query 형태의 요청을 RDF검색을 위한 SPARQL 형태의 질의로 변환하여 검색을 수행하고 검색 결과를 Resource Encoder에게 전달한다. 아래의 Discovery Query의 예제이다.

```
PREFIX umo:<http://keti.re.kr/ubicom/umo#>
<?xml version="1.0" encoding="euc-kr">
<UMODiscoveryQuery xmlns="http://www.keti.re.kr/ubicom/umo/"
version="1.0">
```

```
<UMODevice id="KETI_UMO_ID_0001" name="KETI Mobile device 001"/>
<ResourceDiscovery group="001" type="002" item="*">
  <Restriction name="horizontal resolution" compare="greater then or
equals">1024</Restriction>
  <Restriction name="vertical resolution" compare="greater then or
equals">768</Restriction>
</ResourceDiscovery>
</UMODiscoveryQuery>
```

Query Processor는 위와 같은 XML형태의 Discovery Query를 CIB검색을 위해 다음과 같은 SPARQL로 변환한다.

```
PREFIX umo:<http://keti.re.kr/ubicom/umo#>
PREFIX ns:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?x
WHERE {
  ?x ns:type umo:Monitor .
  ?x umo:Monitor_horizon_resolution ?y.Filter(?y >= \\'1024\\')
  ?x umo:Monitor_vertical_resolution ?z.Filter(?z >= \\'768\\')
```

Query Processor는 위의 SPARQL을 수행하여 최종적으로 CIB에서 검색된 아래의 예제와 같은 RDF형태의 데이터를 Resource Encoder에게 반환한다.

```
<rdf:RDF xmlns:rdf="&rdf;" xmlns:umo="&umo;"
xmlns:rdfs="&rdfs;">
<umo:LocalResource rdf:about="&umo:LocalResource_01"
rdfs:label="LocalResource_01"/>
<umo:UMO rdf:about="&umo:UMO_01" umo:ID="KETI_UMO_ID_001"
umo:annotation="KETI Mobile device is..." rdfs:label="UMO_01">
  <umo:HasLocalDevice rdf:resource="&umo:LocalResource_01"/>
  <umo:communicationOf rdf:resource="&umo:Zigbee_001"/>
</umo:UMO>
<umo:UMPC rdf:about="&umo:UMO_Instance_2"
umo:UMPC_driver_position="e://driver/Umpc.dll"
umo:UMPC_driver_size="10" umo:annotation="UMPC Driver..."
rdfs:label="UMO_Instance_2"/>
<umo:Zigbee rdf:about="&umo:Zigbee_001" rdfs:label="Zigbee_001"/>
<umo:Monitor rdf:about="&umo:monitor_01"
umo:ID="KETI_WALLDISPLAY_0001"
umo:Monitor_horizon_resolution="1024"
umo:Monitor_is_LCD="true"
umo:Monitor_size="32inch"
umo:Monitor_vertica_resolution="768"
umo:annotation="WallDisplay is..."
umo:vender="KETI"
rdfs:label="monitor_01">
  <umo:IsLocalResourceOf rdf:resource="&umo:LocalResource_01"/>
  <umo:hasDriver rdf:resource="&umo:UMO_Instance_2"/>
</umo:Monitor>
</rdf:RDF>
```

3. Manipulation Processor

Manipulation Processor는 U-MO Software Platform의 구성요소들의 CIB 정보 추가, 수정, 삭제 요청을 처리하여 CIB에 대한 추가, 수정, 삭제 작업을 수행한다.

CIB 관련 정보의 추가, 수정, 삭제 등의 작업은 다음과 같은 두 가지 경로를 통해 요청된다.

- Resource Discovery에 의한 공유 리소스 정보 추가
- CIB 정보에 대한 정보 추가, 수정, 삭제 요청

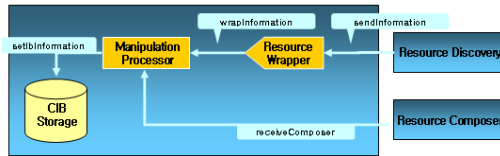


그림 6. Manipulation Processor
Fig. 6. Manipulation Processor

가. Resource Discovery에 의한 공유 리소스 정보 추가

Resource Discovery는 외부의 U-MO들이 제공하는 공유 리소스에 대한 상세 설명인 XML 형태의 Resource Description 메시지를 수신하면 이를 U-MO IB Manager를 통해 CIB에 정보를 저장한다. U-MO IB Manager는 Resource Discovery로부터 Resource Description 정보를 넘겨받아 Resource Wrapper를 통해 일부 정보에 대한 수정 작업을 거친 후 Manipulation Processor를 통해 CIB Storage에 저장하게 된다. Manipulation Processor는 발견된 공유 리소스들에 대한 정보를 U-MO별로 관리하기 위해 CIB storage상에 리소스 정보가 추가될 위치를 결정하고, 결정된 위치에 정보를 추가하는 작업을 수행한다.

Manipulation Processor는 Resource Description 정보로부터 U-MO에 대한 정보를 이용하여 CIB에 해당 U-MO에 대한 정보가 존재하는지 여부를 확인하고 존재하는 경우 Resource Description에 포함된 Resource 정보만을 복사하고, 그렇지 않은 경우 U-MO 정보를 포함하는 전체 Resource Description 정보를 새로이 추가한다. Resource 정보만을 추가하는 경우 Resource의 ID에 의해 중복이 발생하지 않도록 추가 및 수정 작업에 대한 결정 역시 Manipulation Processor에 의해 이루어진다.

나. CIB 정보에 대한 정보 추가, 수정, 삭제 요청
CIB 정보에 대한 Manipulation Processor의 작업은 다음과 같으며 SPARQL 처리를 위한 연산에 대한 추상화된 작업이다.

- 추가 작업 : 추가 작업은 CIB에 구성되어 있는 RDF모델에 새로운 RDF모델을 추가하는 작업을 의미한다. 모델의 추가를 위해 추가할 내용을 포함하는 RDF와 추가될 위치를 지정하는 RDF Subject가 필요하다.
- 수정 작업 : 수정 작업은 CIB에 구성되어 있는 RDF모델을 새로운 RDF모델로 대체하는 작업과 특정 Property나RDF 정보의 내용을 수정하는 작업을 의미한다. 모델을 대체를 위해서는 해당 RDF에 대한 참조 위치를 가지는 RDF Subject와 새로운 내용을 포함하는 RDF 데이터를 가져야 한다.
- 삭제 작업 : 삭제 작업은 CIB에 구성되어 있는 RDF모델에서 일부 모델을 삭제하는 작업을 의미한다. RDF의 삭제를 위해 삭제될 위치를 지정하는 RDF Subject가 필요하다.

4. Resource Encoder

외부 U-MO로부터 공유 리소스에 대한 description 정보 요청 시 U-MO IB Manager는 Query정보 요구 속성에 부합하는 리소스 목록을 검색하고 검색 결과를 이용하여 UMOResourceDescription 정보를 생성하여 이를 반환해야 한다. 이때 Resource Encoder는 Query Processor에 의해 수행된 검색 결과를 이용하여 Resource Provider에게 제공하기 위한 RDF형식의 UMOResourceDescription 정보를 생성하는 역할을 수행한다. Resource Encoder는 Discovery query를 기반으로 Resource 정보를 검색하고, 검색된 Resource 정보와 U-MO 정보를 이용하여 UMOResourceDescription 정보를 생성한다. 생성된 정보는 Resource Provider에 전달되어 서비스 discovery를 요청한 U-MO에게 전송 된다.

5. Resource Wrapper

Resource Wrapper는 Resource Discovery block 으로부터 전달 받은 공유 리소스의 정보를 현재 시스템의 로컬 정보를 기반으로 재구성 하는 다양한 작업을 수행한다. 예를 들어 Resource Description 의 정보 중 해당 리소스의 사용 만료 시간은 사용 가능한 기간을 지정하여 나타낸다. 이 지정된 기간 은 Resource Wrapper에서 현재 시스템의 시스템 시간을 기준으로 재 계산되어 시스템 시간의 형식 으로 변환된다. Resource Description의 정보에서 “ expire=“00:00:30:00” ”와 같이 명시된 30분의 리소 스 사용기간은 Resource Wrapper를 통해 현재 시 스템의 시스템 시간을 기준으로 “ expire=“2007:11:21:12:40:00” ” 와 같이 2007년 11 월 21일 12시 40분 00초에 리소스 사용이 만료됨을 나타내도록 변환된다.

IV. 시나리오

그림 7은 본 논문에서 제안한 CIB프레임워크를 통해 리소스를 공유하는 어플리케이션을 보여주고 있다.



그림 7. 데모 시나리오
Fig. 7. Demo Scenario

사용자가 가지고 있는 하나의 U-MO는 주변의 가능한 리소스를 탐색 후, 자신의 리소스처럼

사용할 수 있다. 이를 통해, 그림 7에서 볼 수 있 듯이 사용자는 자신이 가지고 있는 그림파일을 다른 장치의 리소스를 통해 공유해서 보거나, 보다 큰 화면의 리소스를 통해 편리하게 볼 수 있다.

V. 결론

본 논문에서는 유비쿼터스 모바일 디바이스가 주변 상황을 스스로 인지, 판단하여 서비스를 제공하는 유비쿼터스 지능공간의 구현을 위해 CIB 프레임워크를 제안하였다. CIB 프레임워크는 유 비쿼터스 모바일 디바이스의 리소스 및 정보를 효율적으로 관리하기 위해 RDF 형태로 표현하였 다. RDF를 기반으로 한 이 저장 공간은 새로운 정보의 생성 및 추가, 삭제에 유연하며 저장된 정 보는 SPARQL를 이용하여 질의에 응답한다. CIB 프레임워크는 또한 주변 유비쿼터스 장비간의 정 보 교환이 가능하다. 외부에서 수집된 정보를 자 신의 저장 공간에 맞게 변환하는 기능과 내부 정 보를 외부의 유비쿼터스 모바일 디바이스에 제공 하는 기능을 제공한다.

참고 문헌

- [1] H. Chen et al, Intelligent Agents Meet Semantic Web in a Smart Meeting Room. 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 854-861, July, 2004.
- [2] H. Chen, T. Finin, and A. Joshi. A context broker for building smart meeting rooms. In Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium. 2004 AAI Spring Symposium AAI, March 2004.
- [3] G. Klyne and J.J. Carroll, eds, Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 2004.

- [4] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Candidate Rec. 6 April 2006. <http://www.w3.org/TR/rdf-sparql-query/>.
- [5] World Wide Web Consortium, Extensible Markedup Language (XML) 1.0. W3C Recommendation. See <http://www.w3.org/TR/2006/REC-xml-20060816/>.
- [6] World Wide Web Consortium, RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. See <http://www.w3.org/TR/rdf-schema/>.
- [7] JENA. A Semantic Web Framework for Java, See <http://jena.sourceforge.net/>.

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅및네트워크원천기술개발사업의 08B3-O1-11T 과제로 지원된 것임

— 저 자 소 개 —



이기혁(준회원)
 • 2007년 2월 한국외국어대학교 컴퓨터공학과 졸업
 • 2008년 3월 한국외국어대학교 컴퓨터공학과 석사과정

<주관심분야 : 데이터베이스, 유비쿼터스>



최원철(준회원)
 • 2006년 2월 한국외국어대학교 컴퓨터공학과 졸업
 • 2007년 3월 한국외국어대학교 컴퓨터공학과 석사과정

<주관심분야 : 데이터베이스, 파일시스템>



한지연(준회원)
 • 2005년 3월 한국외국어대학교 컴퓨터공학과 학사과정

<주관심분야 : 알고리즘, 웹, 유비쿼터스>



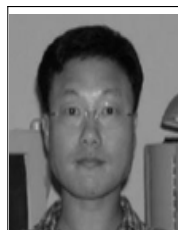
한형진(준회원)
 • 2007년 2월 한국외국어대학교 컴퓨터공학과 졸업
 • 2008년 3월 한국외국어대학교 컴퓨터공학과 석사과정

<주관심분야 : 데이터베이스, 유비쿼터스>



한경훈(준회원)
 • 2002년 3월 한국외국어대학교 컴퓨터공학과 학사과정

<주관심분야 : 데이터베이스, 파일시스템>



손기락(준회원)
 • 1984년 서울대학교 계산통계학과 학사
 • 1986년 서울대학교 계산통계학과 석사
 • 1993년 미국, Univ. of California, Santa Cruz, 전산학 박사

• 1994년 ~ 1996년 전자통신연구원 선임연구원
 • 1996년 ~ 한국외국어대학교 컴퓨터및정보통신공학과 교수

<주관심분야 : 데이터베이스, 데이터마이닝>